

Push Button with STM32F0

In this tutorial, we will learn to use the GPIO pins of the STM32F0 dev board as digital input pins. In the last tutorial, we have seen how to use GPIO pins as digital output pins. For demonstration, we will also show how to interface a push button with the STM32F0 and read the state of a push button. Furthermore, if the push button is pressed LED will turn on, and otherwise LED remains off.

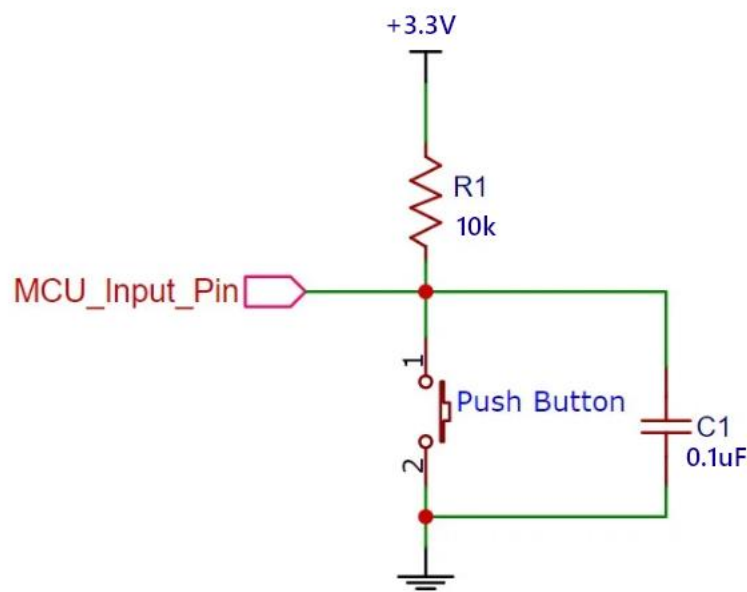
Push Button Interfacing with STM32F0

When we want to interface a push button with a microcontroller, we actually want to read the state of the push button either pressed or not pressed state. When a push button is pressed or unpressed, it provides either logic high or logic low output depending on the configuration mode in which the push button circuit is designed.

There are two configuration modes to interface a push button with the STM32F0 dev board. Let's discuss both these modes one by one.

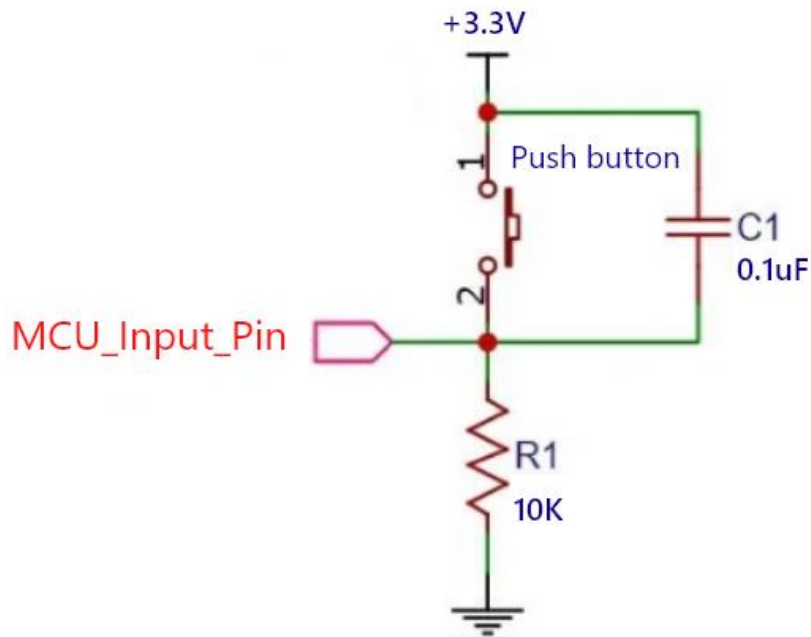
Pull-Up Mode

In Pull-up mode, when a push button is not pressed, a logic high input appears at MCU_Input_Pin. Because a 3.3V signal appears on the input terminal through an R1 resistor. On the contrary, when the push button is pressed, metallic contacts of the push button make contact with the ground terminal and the input terminal. Therefore, a logic low input will reflect on the digital input pin of the STM32F0 dev board. In short, by reading this state of the push button with a digital input pin of a microcontroller, we can identify whether a push button is pressed or not. The following schematic diagram shows the connection of a push button with a pull-up resistor.



Pull-Down Mode

In Pull-down mode, when a push button is not pressed, a logic low input appears on the STM32F0 GPIO pin. Because a ground reference signal appears on the input terminal through an R1 resistor. On the contrary, when the push button is pressed, metallic contacts of the push button make contact with the +3.3V signal and the input terminal. Therefore, a logic high input reflects on the digital input pin of the STM32F0 Dev board. The following schematic diagram shows the connection of a push button with a pull-down resistor.



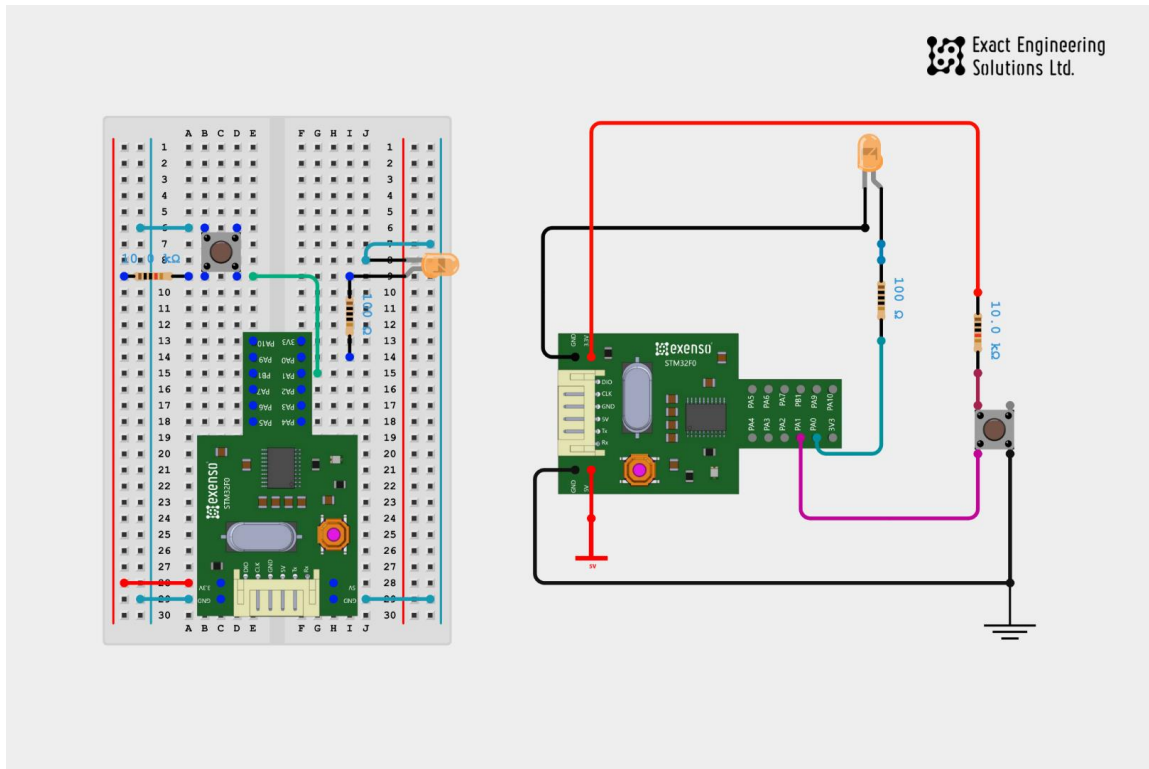
Components Required

You will need the following components –

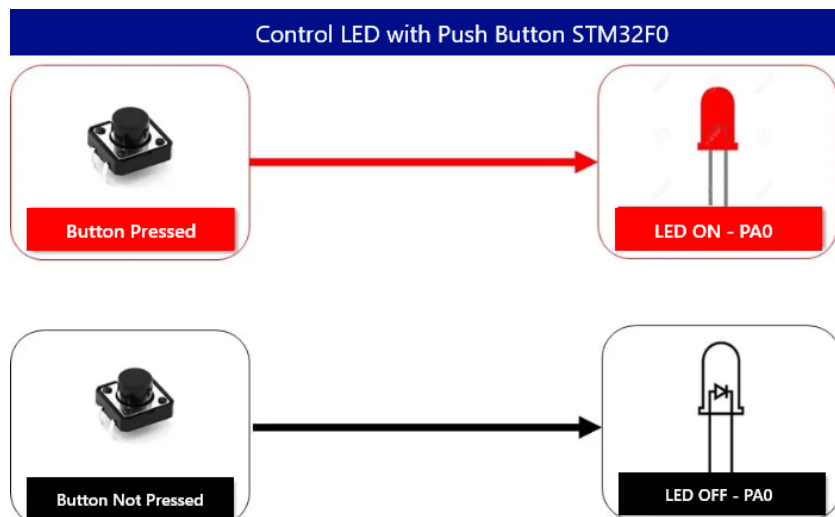
- 1 × Breadboard
- 1 × STM32F030F4P6
- 1 × led
- 1 × push button
- 1 × 100Ω Resistor
- 1 × 10kΩ Resistor
- Some Jumper wire

Procedure

Follow the circuit diagram and hook up the components on the breadboard as shown in the image given below.

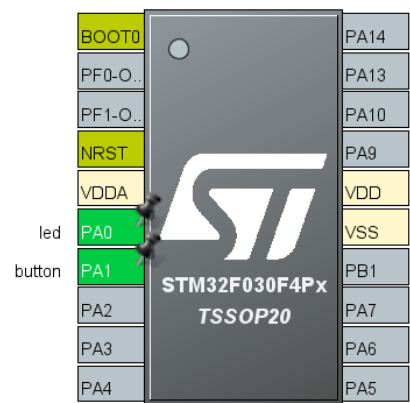
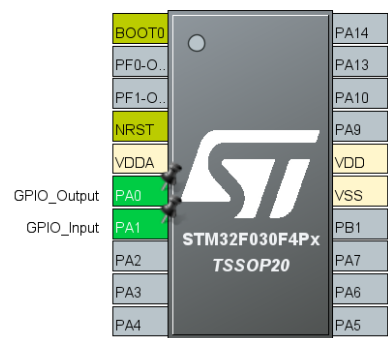


- The PA1 pin will read the output of the push button, if the state is active low, it will turn on the LED of STM32F0. Because the push button is connected with the PA1 pin in pull-up configuration and under normal conditions active high output will appear on the PA1 pin
- When the push button is pressed, the LED will be on, otherwise LED will remain off.



STM32F0 Pin Configuration:

PA0 pin output and PA1 pin Input.



CODE

```
#include "main.h"

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
int main(void)
{
    HAL_Init();

    SystemClock_Config();
    MX_GPIO_Init();

    uint8_t button, button_os;

    button_os = HAL_GPIO_ReadPin(button_GPIO_Port, button_Pin);
    while (1)
    {
        button = HAL_GPIO_ReadPin(button_GPIO_Port, button_Pin);
        if(button != button_os){

            button_os = button;

            if(button == GPIO_PIN_RESET ){
                HAL_GPIO_WritePin(led_GPIO_Port, led_Pin, GPIO_PIN_SET);
            }
            else{
                HAL_GPIO_WritePin(led_GPIO_Port, led_Pin, GPIO_PIN_RESET);
            }
        }
    }
}
```